IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

A Literature Review of Artificial Intelligence and Machine Learning in Digital Security

Priyanka Sushil Ghode[1] and Varad Goraksha Lohakane[2]
[1]Lecturer, Department of Information Technology, Government Polytechnic, Awasari Khurd, Maharashtra, India
[2]Student, Department of Information Technology, Government Polytechnic, Awasari Khurd, Maharashtra, India
lohakanevarad@gmail.com

Abstract: *Modern communication, finance, education, healthcare, and public administration heavily depend on interconnected digital systems, making the protection of computing resources a key engineering task. Traditional security methods that rely on static rules and manually set signatures often struggle to manage the size, speed, and adaptability of today's cyber threats. This paper provides a structured review of the literature on how artificial intelligence and machine learning are used in digital security, the performance levels reported in academic and industry sources, and the factors that limit the reliable deployment of these technologies in real-world settings. Relevant studies were identified through targeted searches and verified using publisher proceedings, academic databases, bibliographic indexes, and official reports, resulting in thirty-two confirmed sources from 2019 to 2026, along with foundational research. The evidence is organized across six main areas: network intrusion detection, malware analysis, phishing and malicious link detection, transaction fraud and user behavior monitoring, log and threat intelligence review, and security for cloud, connected devices, and endpoints. Although many published studies report strong classification results on benchmark datasets and some systems show practical applications, critical evaluations show that experimental outcomes are often exaggerated due to biased data selection and time-related errors. Additionally, low rates of attacks lead to many false alarms, evolving attack tactics cause models to degrade, and learning systems are at risk of being manipulated directly. Issues such as model transparency, data privacy, and analyst fatigue remain unresolved. As a result, learning-based security measures are best used as tools to support traditional security controls and human judgment rather than as fully autonomous replacements*

Keywords: cybersecurity; intrusion detection; malware classification; phishing detection; adversarial machine learning; explainable artificial intelligence

I. INTRODUCTION

Critical societal functions in banking, healthcare, education, trade, and governance now rely on networked computing systems. Ensuring the protection of underlying software, communication channels, data storage, endpoints, and human operators is therefore a fundamental engineering need. Traditional information security practices focus on three core principles: confidentiality, ensuring data is only accessed by authorized users; integrity, preventing unauthorized changes or damage to information; and availability, guaranteeing timely and reliable access to computing resources [1]. Adversarial threats targeting these principles have grown in both size and complexity. Intrusions, malware, deceptive messages, and financial fraud increasingly use automated tools, while attackers continuously refine their tactics to bypass security measures. According to a report from the United Kingdom National Cyber Security Centre (NCSC), malicious actors are already using artificial intelligence (AI) to enhance reconnaissance, vulnerability discovery, and social engineering efforts. These capabilities are expected to increase the frequency and impact of cyberattacks by 2027 [2].

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

Existing security controls like cryptographic hashes, blacklists, and rule-based systems remain essential because they accurately detect known threats and provide clear audit records. However, deterministic systems depend entirely on prior knowledge of threats, and evaluations show that static rules often miss new attack methods and can be bypassed by attackers who understand the rules [3].

These limitations have driven considerable research into AI and machine learning (ML). Within AI, ML involves algorithms that learn patterns from data rather than following instructions written by programmers [4], [5]. Deep learning (DL) builds hierarchical structures through multiple layers of neural networks [6]. Since enterprise security monitoring generates large volumes of network traffic, system logs, executable files, emails, and transaction data, statistical learning is naturally suited for security analysis [7]. Despite this, adopting ML without caution poses significant risks. Early research showed that detecting network anomalies differs significantly from standard pattern recognition due to the high cost of errors and challenges in evaluating performance [8]. A comprehensive audit of methodologies also confirmed that subtle experimental flaws are common in top-tier security publications [9].

The main issue explored in this review is how much published research actually supports using AI and ML in different areas of digital security, what conditions are needed for reported results to hold up, and what practical limits affect reliability in live environments. Since individual studies test different datasets and use varying evaluation metrics, comparing them side by side offers better guidance than looking at isolated test scores. This paper is a narrative literature review that brings together peer-reviewed studies and official reports without running new laboratory experiments. While AI as a field began in the mid-twentieth century [5], integrating these learning models into everyday defensive security operations is a much more recent development.

A. Research Questions

This study focuses on four main questions. First, which statistical and neural learning techniques have been used for specific digital security problems? Second, what detection accuracy do published papers report, and what testing setups produced those numbers? Third, what practical benefits, day-to-day limitations, and security weaknesses appear in the research record? Fourth, which gaps in testing methods and real-world deployment need the most attention from researchers and engineers? Sections V and VI cover the first two questions, Sections VII and VIII look at the third question, and Sections VIII through X address the fourth question.

B. Scope and Engineering Contribution

Most existing review papers look at only one specific area, such as network intrusion detection [10], malware analysis [11], or phishing website detection [12], or they focus on a single broad topic like adversarial attacks [13], explainable artificial intelligence (XAI) [14], or federated learning (FL) [15]. The goal of this paper is to combine these separate topics into one clear reference for computer science students and working engineers. It separates lab test scores from proven real-world systems and distinguishes independent studies from vendor marketing claims. Because IT professionals increasingly manage tools that include ML features, knowing where statistical models work well and where they fall short is essential for building reliable security systems.

II. LITERATURE REVIEW

The thirty-two sources selected for this paper fall into three main groups: broad surveys and conceptual models, original system designs with practical tests, and critical audits of research methods. Table I summarizes the seventeen main comparative studies, and the subsections below discuss their core findings.

A. Surveys and Conceptual Frameworks

Sarker et al. [7] introduce the idea of cybersecurity data science and propose a multi-layer framework for pulling security rules out of system data, providing a conceptual overview rather than new lab experiments. Ferrag et al. [10] review DL models for network intrusion detection, list thirty-five public datasets, and test seven neural network types—recurrent neural networks (RNN), deep neural networks (DNN), restricted Boltzmann machines (RBM), deep

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

belief networks (DBN), convolutional neural networks (CNN), deep Boltzmann machines (DBM), and autoencoders—on two traffic datasets using detection rate, false alarm rate, and overall accuracy. Gibert et al. [11] review ML methods for finding and grouping malware, comparing manually created static and dynamic features with end-to-end neural networks. In a review of eighty phishing detection papers published over five years, Safi and Singh [12] note that supervised ML is the most common approach, with random forest models used in thirty-one studies and a CNN model reaching the highest reported accuracy of 99.98%. Other broad surveys study adversarial attacks and defenses for security models [13], XAI methods across over three hundred security papers [14], and FL setups for privacy-focused intrusion detection [15]. More recently, Xu et al. [16] reviewed 185 studies chosen from more than 40,000 papers on using large language models (LLMs) in cybersecurity, pointing out both their usefulness in finding vulnerabilities and malware and the ongoing need for stronger reliability and clearer explanations.

B. System Designs and Empirical Evaluations

Papers that present original systems show how learning algorithms work on actual security data. Mirsky, Doitshman, Elovici, and Shabtai [17] built Kitsune, an online unsupervised network anomaly detector that uses a group of small autoencoders (KitNET). It reached detection accuracy similar to offline models while running in real time on a single-board Raspberry Pi. Du, Li, Zheng, and Srikumar [18] created DeepLog, which treats system log streams like natural language sequences using a long short-term memory (LSTM) network to spot unusual execution paths and help find the root cause of issues. Veeramachaneni, Arnaldo, Cuesta-Infante, Korrapati, Bassias, and Li [3] built a system that keeps human analysts in the loop by pairing unsupervised outlier detection with a supervised random forest model. Tested on 3.6 billion log lines over three months, this combined setup found 3.41 times more attacks on average and cut false alarms by a factor of five compared to using unsupervised detection alone. Dal Pozzolo, Boracchi, Caelen, Alippi, and Bontempi [19] developed a realistic model for credit card fraud that handles extreme class imbalance, changing customer habits, and delayed verification, testing their incremental learning approach on over 75 million online transactions.

C. Critical and Methodological Audits

A third set of papers checks whether high scores on benchmark tests hold up in real-world conditions. Pendlebury, Pierazzi, Jordaney, Kinder, and Cavallaro [20] showed that Android malware models claiming F1 scores up to 0.99 often rely on unrealistic ratios of safe to malicious apps and time-based data errors. When they tested three models on 129,000 apps gathered over three years using their TESSERACT tool, model accuracy dropped quickly over time. Engelen, Rimmer, and Joosen [21] examined the popular CICIDS2017 intrusion dataset and found major mistakes in how traffic was generated, how network flows were closed, how features were calculated, and how labels were assigned; fixing these errors changed model test results noticeably. Arp et al. [9] listed ten common research mistakes—such as biased sampling, bad labels, data leakage, and poor metric choices—and found them across thirty papers from top security conferences. Earlier work by Axelsson [22] used Bayes' theorem to show that when real attacks are very rare, even a detector with a tiny false-positive rate will produce a flood of false alarms. Alahmadi, Axon, and Martinovic [23] backed this up through interviews with twenty-one security operations center (SOC) analysts. Likewise, Apruzzese et al. [24] looked at industry case studies and found that real-world use of ML in security is still in its early stages and often differs sharply from academic assumptions.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

TABLE I: SUMMARY OF PRINCIPAL REVIEWED STUDIES ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING IN DIGITAL SECURITY

Study & Ref.	Primary Objective	Methodological Approach	Evaluation Corpus / Scope	Key Reported Findings	Principal Limitations
Sarker et al. [7]	Define cybersecurity data science and design a modeling structure	Multi-layer conceptual ML security framework	Conceptual review (no test dataset used)	Organizes security analysis around automatic pattern extraction from system data	Purely theoretical; does not include experimental tests
Ferrag et al. [10]	Review and test DL models for network intrusion detection	Seven neural network types (RNN, DNN, RBM, DBN, CNN, DBM, autoencoder)	Lists 35 datasets; runs tests on CSE-CIC-IDS2018 and Bot-IoT	Measures binary and multi-class detection rates, false alarms, and accuracy on two datasets	Practical testing is limited to just two benchmark datasets
Gibert et al. [11]	Review ML techniques for finding and grouping malware	Compares manual static and dynamic features with neural network features	Literature review (no single dataset)	Outlines feature extraction steps, scaling limits of older ML, and new DL trends	Review paper without new lab experiments
Safi & Singh [12]	Systematically review methods for detecting phishing websites	Blocklists, visual checks, rule heuristics, traditional ML, and DL	80 papers published over a five-year span	Supervised ML appears in 57 papers and random forest in 31; a CNN reached 99.98% accuracy	Scores come from different datasets and cannot be compared directly
Mirsky et al. [17]	Build a lightweight online network anomaly detector	KitNET unsupervised autoencoder group with incremental feature tracking	Kitsune Network Attack Dataset; tested on a Raspberry Pi	Matches offline detection rates while using very little memory and processing power	Spots unusual traffic but cannot name the specific type of attack
Du et al. [18]	Spot and explain execution errors in system logs	LSTM sequence model combined with workflow state tracking	Large-scale production and benchmark log datasets	Beats standard log mining tools in finding anomalies and locating root causes	Needs raw logs to be parsed into structured templates first

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

Study & Ref.	Primary Objective	Methodological Approach	Evaluation Corpus / Scope	Key Reported Findings	Principal Limitations
Veeramachaneni et al. [3]	Combine human analyst feedback with automated attack detection	Unsupervised outlier models paired with a supervised random forest	3.6 billion log lines over 3 months from a large web platform	Boosts detection rates by 3.41 times and cuts false alarms fivefold versus unsupervised models alone	Tested on one platform; missing full ground truth; includes vendor authors
Dal Pozzolo et al. [19]	Model credit card fraud under real-world business limits	Incremental and ensemble learning for class skew, drift, and delayed labels	More than 75 million online card transactions	Handles severe class imbalance, changing user habits, and delayed verification well	Leaves out exact private features and raw error counts in the summary
Pendlebury et al. [20]	Remove sample ratio and time biases when testing malware models	TESSERACT testing tool applied to three Android malware classifiers	129,000 Android apps collected over three years	Proves that F1 scores near 0.99 come from time leakage and unrealistic class ratios	Practical testing only covers Android apps
Engelen et al. [21]	Check data creation and label accuracy in intrusion datasets	Traffic reconstruction, flow fixes, and model re-testing	CICIDS2017 network intrusion dataset	Finds widespread errors in traffic simulation, flow handling, and labels; fixes improve baselines	Audit only looks at one benchmark dataset
Arp et al. [9]	List common research mistakes in security ML papers	Flaw classification, literature audit, and practical case studies	30 papers from top-tier security conferences	Finds ten common errors across data gathering, preparation, testing, and deployment	Audit sample is limited to 30 selected conference papers
Apruzzese et al. [24]	Evaluate how ready ML is for everyday security operations	Critical review paired with two industry case studies	Multi-domain review and practitioner analysis	Shows real-world use is still early and often differs from academic research	Case studies only cover a few specific company workflows
Alahmadi et al. [23]	Study how security analysts view and handle false-positive alerts	Semi-structured interviews with practitioners	21 SOC analysts and security staff	Confirms false alarms strain teams and lists five traits needed for useful alerts	Relatively small interview group

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

Study & Ref.	Primary Objective	Methodological Approach	Evaluation Corpus / Scope	Key Reported Findings	Principal Limitations
Rosenberg et al. [13]	Group adversarial attacks and defense methods in security	Security-focused breakdown of evasion, poisoning, and defense tools	Broad literature review	Shows that modifying real security files for adversarial attacks faces strict technical limits	Survey paper without new lab experiments
Capuano et al. [14]	Review explainable AI techniques used in cybersecurity	Survey of post-hoc and built-in explanation methods	More than 300 reviewed papers	Shows explanations help analysts check alerts but can also reveal model logic to attackers	Literature review without user testing
Agrawal et al. [15]	Review federated learning setups for intrusion detection	Survey of decentralized training methods and privacy tools	Review of federated IDS research	Covers communication overhead, uneven data, and poisoning risks along with possible fixes	Literature review without new experimental tests
Xu et al. [16]	Map how large language models are used across security tasks	Systematic literature review (LLM4Security)	185 papers chosen from over 40,000 candidates	Lists LLM uses in vulnerability and malware analysis while noting reliability and security risks	Covers a fast-changing field; cited from an accepted preprint

Source: Compiled and summarized by the authors from the cited papers; 'no test dataset used' marks a conceptual or review study.

TABLE II: SYNTHESIS OF REVIEWED LITERATURE BY DIGITAL SECURITY APPLICATION DOMAIN

Security Domain	Representative Citations	Primary Learning Methods	Core Reported Capabilities	Persistent Operational Constraints
Network intrusion detection	Ferrag et al. [10]; Mirsky et al. [17]; Engelen et al. [21]	Deep neural networks; online autoencoder groups	High test scores on standard datasets; fast anomaly scoring on small edge devices	Bad labels in public datasets; too many false alarms when attacks are rare; changing network traffic
Malware detection and triage	Gibert et al. [11]; Pendlebury et al. [20]; Microsoft [25]	Static and dynamic feature models; deep neural networks; cloud ensembles	Strong offline test accuracy; vendor-reported stopping of a large coin-mining attack	Biased test setups; fast drop in accuracy over time; attackers modifying files to evade detection

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

Security Domain	Representative Citations	Primary Learning Methods	Core Reported Capabilities	Persistent Operational Constraints
Phishing, spam, and web links	Safi and Singh [12]; Kumaran [26]	Random forest models; convolutional networks; combined text-and-URL filters	Widespread use of supervised ML; high test scores; vendor-reported blocking of 100 million extra daily spam emails	Test scores from different datasets cannot be compared; short-lived domains and hidden text tricks
Transaction fraud and behavior	Dal Pozzolo et al. [19]; Veeramachaneni et al. [3]	Models built for class skew and drift; unsupervised outlier detection with human feedback	Directly handles extreme class imbalance and late verification; analyst input greatly improves precision	Slow confirmation of true fraud; practical tests limited to single company platforms
Threat intelligence and logs	Du et al. [18]; Veeramachaneni et al. [3]; Alahmadi et al. [23]	LSTM sequence models on parsed logs; active learning with human analysts	Automatic flagging of unusual log patterns; structured alert details help analysts review cases faster	Heavy workload on human analysts; complex sequence models are hard to interpret
Cross-cutting methodological issues	Arp et al. [9]; Rosenberg et al. [13]; Capuano et al. [14]; Agrawal et al. [15]; Xu et al. [16]; Apruzzese et al. [24]; Vassilev et al. [27]	Method audits, adversarial taxonomies, and surveys on XAI, FL, and LLMs	Highlights common testing mistakes and organizes risks around evasion, explainability, privacy, and generative AI	Lack of independent, long-term field tests in live production networks

Source: Authors' comparative summary of the cited academic and industry sources.

D. Synthesis of Agreements, Tensions, and Gaps

Looking across the reviewed papers, clear areas of agreement and disagreement stand out, as shown by domain in Table II. Researchers generally agree that statistical and deep learning models can spot complex patterns across huge streams of monitoring data that human rule-writers would miss [3], [7], [18], and that poor experimental design is the biggest obstacle to reliable results [9], [20], [21], [24]. On the other hand, there is a sharp divide between survey papers that highlight near-perfect test results—such as 99.98% accuracy in phishing detection [12]—and critical audits showing that model accuracy drops sharply once sample ratio and time biases are fixed [20]. These differences come from the gap between static offline test splits and live production networks, where real attacks are rare, verified labels arrive late, and attackers constantly adapt [19], [20], [22]. Because of this, getting a high score on a benchmark dataset does not guarantee a model will work reliably in practice, and accuracy percentages cannot be compared directly across different datasets.

III. REVIEW OBJECTIVES

To answer the research questions laid out in Section I, this review follows six main objectives:

1. Explain the basic concepts of AI, ML, DL, and digital security (Sections V-A and V-B).
2. Review how learning algorithms are used across six key areas of cybersecurity (Sections V-C through V-H).

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 6, October 2026 Impact Factor: 8.2

3. Assess the real-world performance reported in academic prototypes and commercial tools (Sections V-C through V-I).
4. Compare the practical pros and cons of traditional rule-based systems and learning-based methods (Sections VI and VII).
5. Point out common testing flaws, adversarial weaknesses, and open research problems (Section VIII).
6. Provide practical engineering guidelines for security researchers and organizations deploying these tools (Sections VIII and X).

IV. RESEARCH METHODOLOGY

A. Review Design and Information Sources

This paper uses a structured narrative review approach to compare modeling assumptions, testing methods, and practical limits across representative cybersecurity studies. Unlike a statistical meta-analysis, it does not combine numerical results from different studies into a single score. Relevant papers were found through targeted searches across peer-reviewed publisher sites and conference records (including IEEE, Association for Computing Machinery (ACM), Springer, Elsevier, USENIX, and the Network and Distributed System Security Symposium (NDSS)), open-access archives (such as arXiv and university repositories), bibliographic indexes (dblp and OpenAlex), and official government standards and reports from the National Institute of Standards and Technology (NIST) and the UK NCSC. The literature search and a two-step reference check were completed in October 2026, confirming publication details and Digital Object Identifier (DOI) records for all thirty-two selected sources.

B. Search Strategy, Screening Criteria, and Assessment

Search terms combined security topics with learning methods, including phrases such as artificial intelligence in cybersecurity, machine learning intrusion detection, deep learning malware classification, phishing webpage detection, cybersecurity threat intelligence, adversarial machine learning, explainable artificial intelligence in security, federated learning intrusion detection, and credit card fraud concept drift. We focused mainly on peer-reviewed papers and official agency reports published between 2019 and 2026, along with foundational studies that introduced core concepts or landmark systems [3], [8], [22]. Papers were kept only if they focused on digital security tasks, had verifiable publication records, and explained their methods clearly enough to be evaluated critically. Unverified web articles and marketing pages were left out; two technical reports from commercial vendors [25], [26] are included only as examples of industry deployment and are clearly marked as company-reported claims. Each selected paper was checked for publication quality, dataset realism, independence from software vendors, and honesty about limitations, and reported test scores are always discussed alongside the specific dataset used.

V. DOMAIN FINDINGS AND DOCUMENTED IMPLEMENTATIONS

This section brings together the technical findings across major areas of digital security. Subsections V-A and V-B explain the basic concepts, Subsections V-C through V-H look at specific security tasks, and Subsection V-I reviews documented real-world systems.

A. Fundamentals of Artificial Intelligence and Machine Learning

In computer science, AI refers to building computer systems that can sense their environment, reason with information, and perform tasks that normally require human intelligence [5]. Within AI, ML covers algorithms that get better at a task by learning patterns from training data [4], while DL uses neural networks with multiple layers to automatically learn layered features directly from raw inputs [6], [28]. As shown in Fig. 1, AI is a broader field than ML: rule-based expert systems, symbolic logic, and heuristic search are classic branches of AI that do not learn from data.

Security analysis relies mostly on two main learning approaches, which are compared in Table III. In supervised learning, models learn a decision boundary from labeled examples of safe and malicious activity so they can classify new inputs. In unsupervised learning, algorithms look at unlabeled data to learn what normal behavior looks like and flag unusual patterns; anomaly detection, which means finding data points that differ noticeably from expected

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

behavior [29], is its main use in security. Reinforcement learning, where an agent learns actions through trial-and-error rewards, plays only a small role in the empirical studies reviewed here. Hybrid systems blend both methods by sending unusual events found by unsupervised models to human analysts, whose decisions are then used to train a supervised classifier [3].

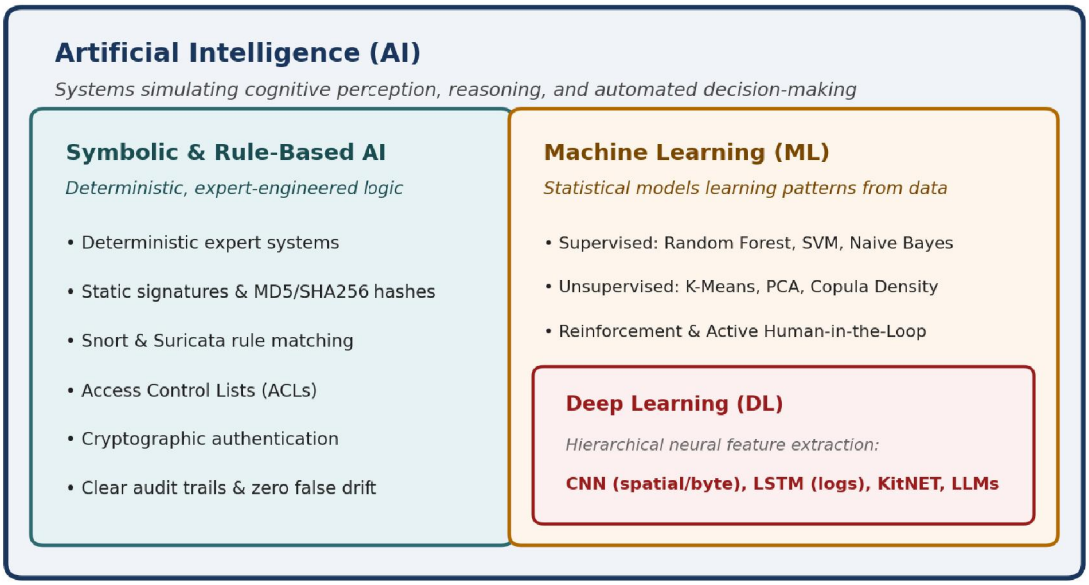

Fig. 1. Hierarchical relationship among artificial intelligence, machine learning, deep learning, and symbolic rule-based systems.

TABLE III: COMPARISON OF SUPERVISED AND UNSUPERVISED LEARNING PARADIGMS IN DIGITAL SECURITY

Analytical Dimension	Supervised Learning	Unsupervised Anomaly Detection
Training data requirement	Labeled datasets with confirmed safe and malicious examples	Unlabeled system data that is assumed to be mostly normal activity
Core modeling mechanism	Learns a mathematical function that maps input features to known threat types	Models normal baseline behavior and scores how far new events deviate
Primary security tasks	Grouping malware families, filtering phishing and spam, spotting variations of known threats	Spotting new network anomalies, unusual system log patterns, and novel account misuse
Operational strengths	Predicts specific threat types directly; easy to test offline	Works without labeled attack records; can flag new and previously unseen patterns
Operational limitations	Getting good labels is costly and labels quickly go out of date; misses completely new attack types	Unusual activity is not always an attack, which causes frequent false alarms [3]; does not name the attack type
Documented examples	Random forest and CNN phishing filters [12]	Kitsune autoencoder group [17]; outlier detection stage in AI2 [3]

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

Note: Hybrid systems pair unsupervised outlier scoring with supervised classification trained on labels confirmed by human analysts.

Common supervised models include decision trees, random forests, support vector machines (SVMs), logistic regression, and multi-layer neural networks. Unsupervised security tools often use clustering, principal component analysis (PCA), and neural autoencoders trained to rebuild normal inputs so that a high reconstruction error points to an anomaly [17]. Sequence models like LSTMs track time-based patterns in system logs [18], while CNNs pick up local spatial or character-level patterns in phishing links and malware files [11], [12].

Model predictions are measured by counting true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Table IV lists the five standard evaluation metrics and shows why overall accuracy can be misleading when real attacks make up only a tiny fraction of daily traffic [22].

TABLE IV: STANDARD EVALUATION METRICS FOR DIGITAL SECURITY DETECTION SYSTEMS

Evaluation Metric	Mathematical Formula	Operational Significance in Security
Precision	TP / (TP + FP)	Share of triggered alerts that are real attacks; directly affects how much time analysts spend checking false alarms
Recall (detection rate)	TP / (TP + FN)	Share of actual attacks that the system catches; low recall means real intrusions slip through
False positive rate	FP / (FP + TN)	Share of normal events wrongly flagged as attacks; even a tiny rate creates thousands of false alerts when normal traffic is huge [22]
F1 score	2 × Precision × Recall / (Precision + Recall)	Balances precision and recall into one number, but hides whether mistakes come from missed attacks or false alarms
Accuracy	(TP + TN) / (TP + FP + TN + FN)	Total share of correct predictions; very misleading when safe events vastly outnumber attacks

Source: Standard statistical formulas paired with practical security interpretations from the reviewed literature.

B. Core Principles of Digital Security

Digital security—often called cybersecurity—covers the technical and administrative safeguards used to protect computer hardware, software, networks, data, and users from unauthorized access, disruption, changes, or destruction based on confidentiality, integrity, and availability [1]. Security monitoring usually combines signature detection, which looks for known signs of an attack, with anomaly detection, which looks for unusual changes from normal behavior. Fig. 2 shows how the six main security areas covered in this paper connect to their data sources and common learning models.

Threats and vulnerabilities: A threat is any possible event or actor that can cause harm, while a vulnerability is a software bug, design flaw, or setup mistake that an attacker can use to break in.

Malware: Malicious software—such as ransomware, trojans, spyware, and hidden crypto-miners—is built to harm system confidentiality, integrity, or availability.

Phishing: Fake emails, messages, and look-alike websites pretend to be trusted organizations to steal login details or financial data from users [12].

Network intrusion: Unauthorized movement or tampering across network and host systems is tracked by an intrusion detection system (IDS).

Authentication and access control: Authentication checks who a user or device is, while access control limits each user to only the permissions they need.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

Detection and response: The NIST Cybersecurity Framework (CSF) 2.0 groups security tasks into six main functions: Govern, Identify, Protect, Detect, Respond, and Recover [30]. Learning models are mainly used to help with the Detect function and to support human decisions during the Respond phase.

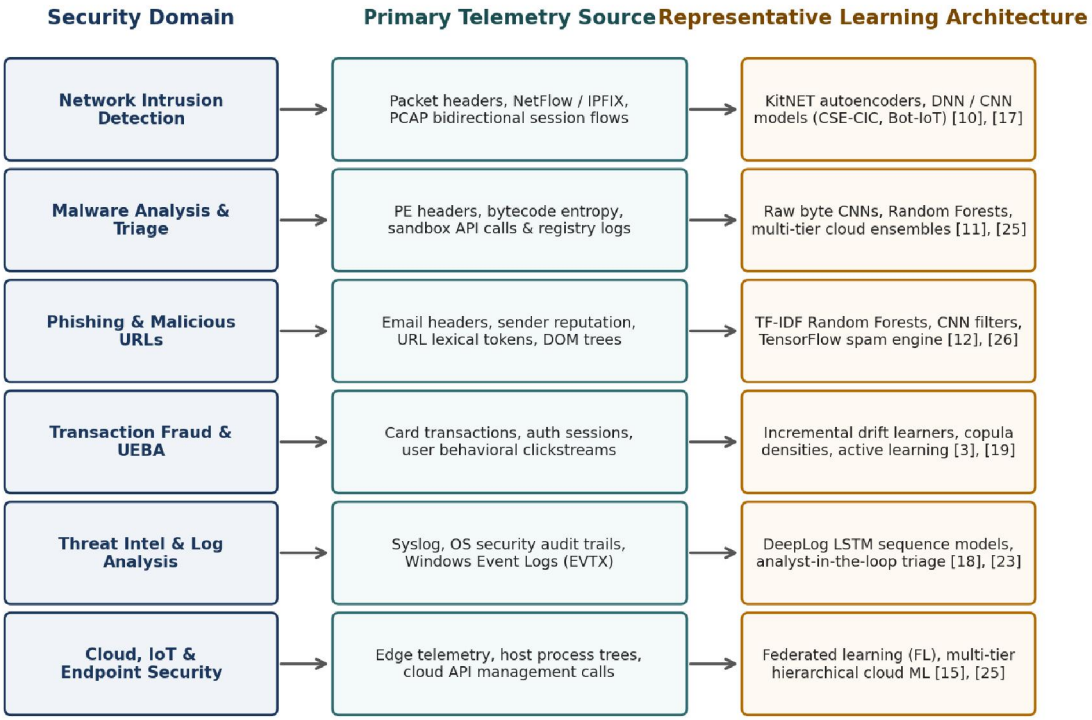

Fig. 2. Mapping of digital security domains to primary telemetry sources and representative learning approaches reported in the reviewed studies.

C. Network Intrusion Detection and Traffic Monitoring

An IDS watches network packets or host activity to spot unauthorized access. Learning-based intrusion tools turn raw packet headers or two-way traffic flows into numerical features, then use supervised models to label known attacks or unsupervised models to score unusual behavior. Fig. 3 shows the step-by-step workflow of a learning-assisted threat detection system, highlighting the important line between a model's statistical prediction (Stage 5) and an actual defensive action (Stage 6). A model only gives a probability score or a label; deciding whether to isolate a computer, block traffic, or open an incident remains a policy choice made by human analysts and surrounding security systems.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

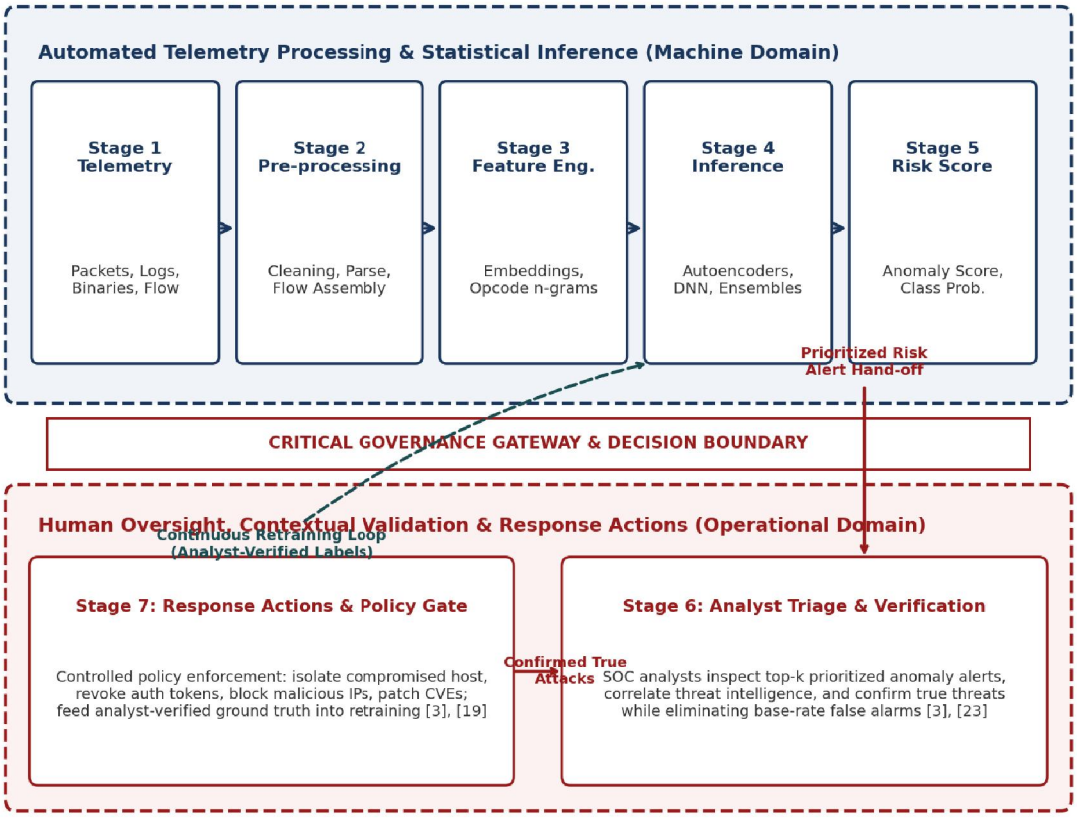

Fig. 3. Operational pipeline of a learning-assisted threat detection system distinguishing statistical model predictions from human-governed response actions.

Practical studies show both the promise and the limits of network intrusion models. Ferrag, Maglaras, Moschoyiannis, and Janicke [10] tested seven deep neural network models on the CSE-CIC-IDS2018 and Bot-IoT traffic datasets, while Mirsky et al. [17] showed that Kitsune's online group of small autoencoders can match the accuracy of offline models even when running on small edge devices. Still, benchmark scores need to be read carefully. Engelen, Rimmer, and Joosen [21] found widespread traffic-generation and flow-labeling mistakes in the CICIDS2017 dataset, and Axelsson [22] proved that when real attacks are rare, false alarms easily outnumber true detections. Because real attacks usually make up less than 0.1% of company log events [3], heavy class imbalance, shifting network traffic, and the need to process data at high network speeds remain major hurdles to deployment.

D. Malware Detection and Family Classification

Malware analysis tools check executable files, mobile apps, or running processes to decide whether they are safe or malicious and to sort malware into known families. Static analysis pulls features from a file without running it, looking at raw byte patterns, disassembled instruction codes, file headers, or requested app permissions. Dynamic analysis runs the file inside a safe sandbox to record what it does—such as application programming interface (API) calls, registry changes, and network connections—while endpoint monitoring watches live processes on user machines. Gibert, Mateu, and Planes [11] review these feature types and show how the field has shifted from manual feature engineering toward deep neural networks that read raw byte sequences or grayscale byte images directly, as shown in Fig. 4(a).

High test scores in a lab do not guarantee lasting protection in the real world. Pendlebury et al. [20] showed that Android malware models reporting F1 scores up to 0.99 benefit from sample ratio bias (unrealistic mixes of safe and

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

malicious apps) and time leakage (training on newer files to test on older ones). When tested in strict time order using TESSERACT, model accuracy drops steadily unless the system is regularly retrained or updated with active learning. In addition, attackers routinely disguise code and tweak unused parts of files to slip past both static and dynamic detectors [13].

E. Phishing, Spam, and Malicious Web Link Detection

Filters for deceptive messages and uniform resource locators (URLs) check email headers, message text, domain registration details, URL wording, webpage code, and page appearance to catch impersonation attacks. In their review of eighty phishing detection papers, Safi and Singh [12] grouped defenses into blocklists, visual comparison tools, rule heuristics, traditional ML, and DL models. Supervised ML was used in fifty-seven papers, random forests appeared in thirty-one papers, and a CNN model reached the highest reported test score of 99.98%. Because these papers test on different static datasets, their accuracy numbers cannot be ranked directly against each other. Fig. 4(b) shows the standard workflow for classifying phishing messages and malicious links.

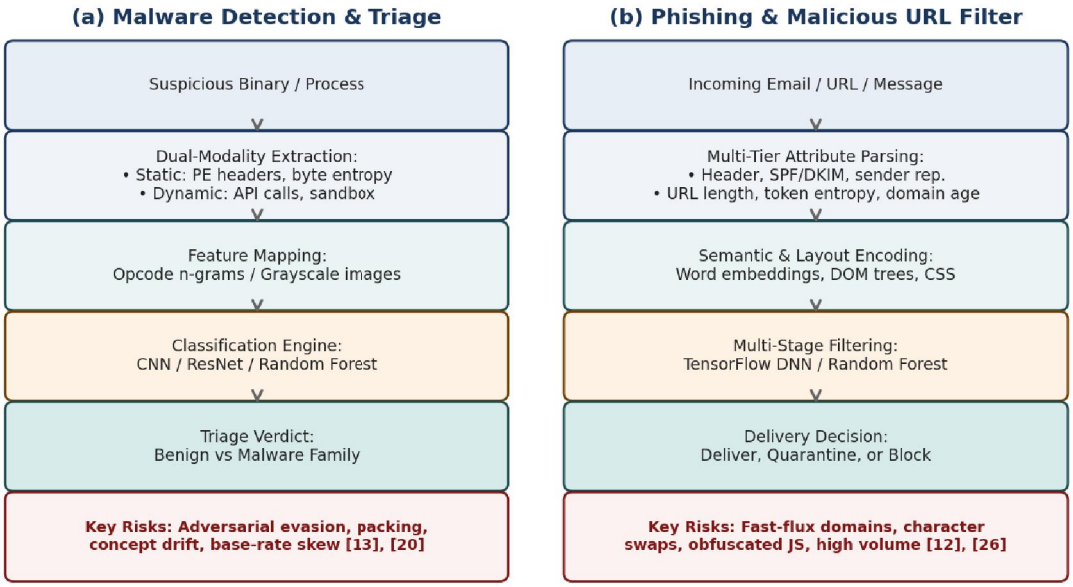

Fig. 4. Comparative detection workflows and operational risks for (a) malware classification and (b) phishing, spam, and malicious web link filtering.

On a commercial level, Google states that Gmail's filtering system blocks over 99.9% of spam, phishing, and malware, and that adding TensorFlow ML models helped block roughly 100 million more spam emails each day by catching image spam, hidden text, and newly created domains [26]. Because this number comes from a company blog post rather than an independent study, it should be viewed as a vendor-reported result. While learning filters scale well and adjust to user habits, they still struggle with short-lived phishing sites, small text changes designed to trick filters [13], and severe class imbalance.

F. Transaction Fraud and User Behavior Analytics

Fraud and account-abuse detectors monitor payment records and user login sessions to spot unauthorized activity. Dal Pozzolo et al. [19] point out that standard offline ML setups fail in live credit card systems because real fraud is very rare, normal shopping habits change with the seasons, and confirmed fraud labels take weeks to arrive after customer disputes are checked. Their learning system separates immediate alert scoring from delayed label updates and was tested on more than 75 million online card transactions. For web security, Veeramachaneni et al. [3] track twenty-four daily behavior metrics per user to catch account takeovers, bot sign-ups, and service abuse.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

The hardest part of monitoring user behavior is telling normal human changes apart from a real attack. Legitimate travel, buying a new phone, or holiday shopping can change a user's normal pattern without any malicious activity, and Veeramachaneni et al. [3] stress that just because an event is statistically rare does not mean it is an attack. While the reviewed studies offer solid evidence on transaction and session monitoring, they do not test biometric identity checks.

G. Threat Intelligence, Log Analysis, and Vulnerability Triage

Company security teams collect millions of operating system logs, firewall entries, and intrusion alerts every day—far more than human staff can read by hand. Sarker et al. [7] describe this task as using data science to automatically pull out rules and patterns that help security teams make decisions. Du et al. [18] show with DeepLog that turning raw log text into structured keys and training an LSTM sequence model makes it possible to spot unusual execution paths and rebuild workflows to find the root cause of problems. Similarly, Veeramachaneni et al. [3] show that giving human analysts a fixed daily list of the top-scoring anomalies and feeding their feedback into a supervised model cuts false alarms sharply. Interview findings by Alahmadi et al. [23] support this focus on human analysts: SOC staff dealing with 99% false-positive rates need alerts that are dependable, clear, analytical, rich in context, and easy to share.

As a result, log and threat-intelligence models work best as tools that help human analysts sort through data faster rather than as fully independent decision-makers. As for software vulnerabilities, while the UK NCSC [2] notes that attackers are using AI to help find flaws and Xu et al. [16] list LLM studies on spotting bugs in source code, independent field tests proving how much these tools help defenders in practice are still rare.

H. Cloud Platforms, Connected Devices, and Endpoint Protection

Modern security systems spread learning tasks across user devices, edge networks, and cloud servers. For endpoint security, Microsoft describes a multi-step setup where local sensors on a computer send suspicious process data to cloud ML models—combining file metadata, static features, and sandbox test models—to block new malware in milliseconds [25]. For Internet of Things (IoT) and embedded devices with limited memory and battery power, Mirsky et al. [17] built Kitsune's small autoencoder group to run directly on edge gateways, and Ferrag et al. [10] tested deep learning models on the Bot-IoT dataset. Because sending raw network logs from many IoT or company devices to one central server uses heavy bandwidth and creates privacy risks, Agrawal et al. [15] review FL methods that share local model updates instead of raw data, while noting ongoing issues with uneven data and model poisoning. Although cloud servers provide the computing power needed for large-scale security analysis [25], [26], independent peer-reviewed studies testing ML specifically for protecting cloud management systems remain scarce in the reviewed literature.

I. Documented Research Prototypes and Commercial Deployments

Table V summarizes six documented systems, clearly separating peer-reviewed academic prototypes tested on real or realistic data from commercial tools described by software vendors. None of the six systems went through an independent third-party audit in the sources we reviewed: academic prototypes were tested by their own creators (sometimes working with industry partners), while vendor blog posts leave out full error tables and code needed to repeat the tests. Taken together, these examples show that ML pipelines can run at large company scale, while highlighting the need for independent real-world testing.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

TABLE V: DOCUMENTED RESEARCH PROTOTYPES AND COMMERCIAL SECURITY DEPLOYMENTS

System & Organization	Target Security Problem	Learning Architecture & Workflow	Documented Empirical Outcomes	Evidence Category	Reported Constraints
AI2 (MIT CSAIL & PatternEx; [3])	Large-scale account takeover, payment fraud, and service abuse	Unsupervised outlier models (PCA, replicator neural net, copula) paired with analyst labels and random forest	Tested on 3.6 billion log lines over 3 months; 3.41× higher detection rate and 5× fewer false alarms than unsupervised alone	Peer-reviewed conference paper; tested on company logs	Data from one platform; incomplete ground truth; includes vendor co-authors
Kitsune (Ben-Gurion Univ.; [17])	Real-time network anomaly detection on small edge devices	KitNET online ensemble of small autoencoders using incremental statistics	Matches offline anomaly detectors across several attack types while running live on a Raspberry Pi	Peer-reviewed conference paper; academic research prototype	Unsupervised anomaly scores do not identify the exact type of attack
DeepLog (Univ. of Utah; [18])	Spotting and diagnosing execution errors in system logs	LSTM sequence model trained on parsed log keys and parameter values with workflow graphs	Outperforms traditional log-mining methods on large log datasets and helps isolate root causes	Peer-reviewed conference paper; academic research prototype	Depends on accurate log parsing templates before modeling
Card-fraud framework (Dal Pozzolo et al. [19])	Online credit card fraud detection	Incremental and ensemble learning separating immediate alert ranking from delayed verification labels	Tested on over 75 million card transactions with severe class skew, drift, and delayed feedback	Peer-reviewed journal paper written with industry partner	Exact private feature details and raw error counts are withheld
Gmail spam/phishing filter (Google; [26])	Email spam, phishing messages, and malicious attachments	Large-scale ML models including TensorFlow classifiers working alongside rule-based filters	Vendor reports blocking over 99.9% of spam/phishing, plus 100M extra daily spam emails	Vendor engineering blog post; no independent audit	False-positive and false-negative rates withheld; private data cannot be re-tested

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 6, October 2026 Impact Factor: 8.2

System & Organization	Target Security Problem	Learning Architecture & Workflow	Documented Empirical Outcomes	Evidence Category	Reported Constraints
Windows Defender Antivirus (Microsoft [25])	Endpoint crypto-mining malware attack (Dofoil outbreak)	Local behavior monitoring connected to cloud ML models (metadata, sample, and sandbox models)	Vendor reports stopping over 400,000 infection attempts within 12 hours with millisecond cloud blocking	Vendor security blog post; no independent audit	Reports count of blocked attempts rather than verified precision or recall

Source: Compiled by the authors from the cited peer-reviewed papers and vendor technical reports.

VI. COMPARATIVE DISCUSSION AND CRITICAL ANALYSIS

A. Conventional Controls Versus Learning-Based Defenses

Table VI compares rule-based security controls with AI and ML methods across seven practical areas. Published research does not support replacing traditional controls with statistical models: cryptographic hashes, access control lists, and static rules remain the most dependable, clear, and low-cost defense against known threats, while learning models add value by spotting subtle behavioral changes and sorting through huge volumes of log data [3], [7].

TABLE VI: CONCEPTUAL AND OPERATIONAL COMPARISON OF CONVENTIONAL AND LEARNING-ASSISTED SECURITY CONTROLS

Engineering Dimension	Conventional Controls (Signatures, Rules, Blocklists)	Learning-Assisted Security Controls	Operational Qualification
Known threat indicators	Matches exact file hashes and static rules with very high accuracy	Spots statistical similarity to previously labeled threat groups	Work best together; static signatures remain better for exact known indicators
Novel or mutated threats	Misses new threats until human analysts write updated rules or signatures [3]	Flags unusual behavior or modified files, though catching every attack is never guaranteed	Model accuracy drops over time as software and attacks change [20]
Decision transparency	Analysts can immediately see which rule or indicator triggered an alert	Internal weights in deep neural networks and large ensembles are often hard to understand	Explainable AI helps clarify decisions but creates new security trade-offs [14]
Ongoing maintenance	Needs human experts to keep writing and updating rule lists	Needs data pipelines, accurate labels, drift checks, and regular model retraining	Engineering work shifts from writing rules to managing data and models
False-alarm behavior	Controlled by how specific the rules and thresholds are	Controlled by training data quality, score cutoffs, and how rare real attacks are	Either approach can flood SOC analysts with alerts if poorly tuned [23]

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

Engineering Dimension	Conventional Controls (Signatures, Rules, Blocklists)	Learning-Assisted Security Controls	Operational Qualification
Adversarial exposure	Bypassed when attackers change the specific details targeted by known rules	Vulnerable to evasion tricks, poisoned training data, and model stealing [13]	Learning systems add a new attack surface that must be protected [27]
Compute and data overhead	Uses very little memory, storage, and processing power	Needs significant computing power for training and feature extraction, though small models exist	Lightweight online models like Kitsune cut down resource use on edge devices [17]

Source: Authors' summary of the reviewed empirical and critical studies.

B. Suitability of Learning Paradigms and Architectures

No single learning algorithm is the best choice for every cybersecurity problem. Supervised models work very well when plenty of reliable, labeled examples are available—such as sorting phishing websites [12] or grouping known malware families [11]—but lose accuracy as attackers change their methods [20]. Unsupervised anomaly detectors do not need labeled attack history and can catch new patterns in network traffic or system logs [17], [18], but they trigger many false alarms because harmless operational changes happen all the time [3]. Systems that keep human analysts in the loop by pairing unsupervised anomaly scoring with supervised models trained on analyst feedback help balance these pros and cons [3]. Likewise, while deep neural networks handle complex sequence data or raw binary files well [11], [18], simpler tree models like random forests often perform just as well on structured table features while using less computing power and being easier to interpret [12].

C. Evaluation Metrics, Base Rates, and Operational Costs

Headline test scores cannot be compared across different papers without looking at class ratios and real-world operating limits. Getting 99.98% accuracy on a balanced phishing test set [12], a 0.99 F1 score on a skewed Android dataset [20], and an 86.8% detection rate within a daily limit of 200 analyst reviews [3] represent completely different practical situations. As Axelsson [22] showed, when real attacks make up only a tiny fraction of all events, even a 1% false-positive rate creates dozens or hundreds of false alarms for every real intrusion. Because of this, practical testing should focus on precision and recall within a fixed daily alert budget rather than overall accuracy, and score cutoffs must balance the risk of missing an attack against the cost of analyst burnout.

D. Data Quality, Temporal Drift, and Generalization

Statistical models can only catch future threats if their training data accurately reflects real production networks. Audits by Pendlebury et al. [20], Engelen et al. [21], and Arp et al. [9] show that fake traffic flaws, bad labels, unrealistic class ratios, and random cross-validation splits (which mix newer and older data together) artificially boost published test scores. In live networks, routine software updates, changing user habits, and adapting attackers cause ongoing concept drift, while confirmed ground-truth labels often take days or weeks to arrive [19]. Moreover, while small edge detectors like Kitsune can run on basic hardware [17], large behavior-monitoring platforms like AI2 need distributed big-data servers capable of scoring tens of millions of records every day alongside ongoing labeling and retraining workflows [3].

E. Explainability, Privacy, Bias, and Risk Governance

Black-box neural network predictions slow down incident response because SOC analysts need to check why an alert fired before taking disruptive steps like blocking a server [23]. While XAI tools can highlight which features drove a prediction to help analysts check alerts, Capuano et al. [14] point out that showing these explanations can also help attackers figure out how the model works and bypass it. Gathering raw network and device logs across different organizations also creates legal and privacy concerns; FL reduces raw data sharing by exchanging local model updates

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 6, October 2026 Impact Factor: 8.2

instead, though decentralized training can still suffer from data leakage and poisoning attacks [15]. In the security papers reviewed here, bias mainly refers to unrepresentative sampling in training datasets [9], [20] rather than demographic unfairness in user profiling. To manage these technical and organizational risks in a structured way, the NIST Artificial Intelligence Risk Management Framework (AI RMF 1.0) sets out four main functions: Map, Measure, Manage, and Govern [31].

F. Adversarial Manipulation and Automation Over-Reliance

Unlike random system noise, cyber attackers deliberately try to trick model inputs and corrupt training pipelines. Over more than ten years of research into adversarial ML [32], studies have shown how evasion attacks slightly alter malware files or network traffic at test time, poisoning attacks slip bad examples into training data, and privacy attacks pull sensitive records out of trained models [13]. The 2025 NIST adversarial ML taxonomy [27] organizes these attack methods across both predictive and generative AI systems and notes that current mathematical defenses offer only partial protection against attackers who adapt their strategy. Making matters worse is over-reliance on automation: treating a model's probability score as a guaranteed trigger for automatic blocking can cause major service outages from false alarms or targeted denial-of-service tricks, which reinforces the need for human oversight shown in Fig. 3 [3], [23].

VII. ADVANTAGES AND DISADVANTAGES OF LEARNING-BASED SECURITY

Reviewing the research record shows that every practical benefit of using AI and ML in cybersecurity comes with a matching technical or operational limitation, as summarized in Table VII.

A. Operational Advantages

High-volume pattern recognition: Statistical and deep learning models can process millions of log entries, network flows, and files that are far too large for humans to check manually [7], [18].

Detection of mutated and behavioral threats: Unsupervised and feature-learning models can spot modified files and unusual behavior patterns that slip past static signatures [3], [17], [18].

Automated alert prioritization: Ranking unusual events into a manageable daily list for analysts [3] and filtering out bulk spam [26] lets security teams focus on the most serious threats.

Execution speed and deployment flexibility: Small online autoencoders can run in real time on edge devices [17], while cloud model ensembles can check suspicious files in milliseconds [25].

Iterative adaptation via analyst feedback: Workflows that keep humans in the loop use analyst decisions to continually update models as new attack patterns appear [3], [18].

B. Technical and Practical Disadvantages

Burdensome false positives and false negatives: When real attacks are rare, even a tiny false-positive rate creates a flood of false alarms [22], [23], while raising the alert threshold increases the risk of missing real attacks.

Sensitivity to dataset quality: Biased samples, artificial test flaws, and wrong ground-truth labels badly skew how models learn and how they are scored [9], [20], [21].

Vulnerability to concept drift: Model accuracy drops over time as normal traffic changes and attackers adjust their methods [19], [20].

Susceptibility to adversarial attacks: Attackers can bypass models by making small allowed changes to inputs, poisoning training data, or probing how the model works [13], [27].

Limited interpretability: Complex neural networks and large ensembles hide how they reach decisions, which slows down analyst checks and incident response [14], [23].

Infrastructure and maintenance overhead: Running models in production requires ongoing data collection, label verification, drift tracking, and regular retraining [3], [19].

Data privacy exposure: Training models on company logs, emails, and user activity creates privacy risks that decentralized learning only partly solves [15].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

TABLE VII: SUMMARY OF CONDITIONAL ADVANTAGES AND CORRESPONDING LIMITATIONS OF LEARNING IN DIGITAL SECURITY

Operational Dimension	Conditional Advantage	Corresponding Disadvantage or Requirement
Telemetry scale	Finds complex patterns across huge volumes of logs and network traffic [7]	Relies heavily on having representative and accurately labeled training data [9], [21]
Novel or modified attacks	Spots unusual behavior and modified files that static rules miss [17]	Catching new threats is never guaranteed, and accuracy drops over time due to drift [20]
SOC alert triage	Ranks a manageable daily list of high-risk events for analysts to check [3]	Low attack rates mean false alarms will dominate if score cutoffs are too loose [22], [23]
Response speed	Supports live scoring on edge devices [17] and fast cloud checks [25]	Model scores are predictions, not actions; automatic blocking can disrupt normal work (Fig. 3)
Lifecycle adaptation	Uses analyst feedback and incremental updates to keep up with changing threats [3], [18]	Requires ongoing drift monitoring, handling of delayed labels, and regular retraining [19]
Decision transparency	Post-hoc XAI tools highlight which features influenced an alert for analysts [14]	Deep neural networks remain hard to explain, and explanations can leak model logic [14]
Pipeline security	Allows multiple models to cross-check predictions	Opens up new attack vectors through evasion, poisoning, and model extraction [13], [27]
Data confidentiality	Federated learning avoids sending raw company or IoT data to a central server [15]	Privacy risks in behavioral data and poisoning threats in decentralized training remain open issues [15]

Source: Authors' summary of the reviewed empirical and critical literature.

VIII. LIMITATIONS, OPEN CHALLENGES, AND FUTURE DIRECTIONS

A. Methodological Limitations of This Review

Because this paper is a structured narrative review rather than a full systematic review, some relevant studies—especially newer papers from 2025 and 2026—may have been left out. In addition, several numerical results were gathered from verified published papers, abstracts, and survey tables rather than by re-running the original source code ourselves, and the two commercial examples [25], [26] rely on numbers reported by the vendors.

B. Principal Research and Deployment Challenges

Separating well-known engineering hurdles from newer research topics helps clarify where the field stands today:

• Adversarial robustness (established challenge): Evasion, poisoning, and extraction attacks against security models are well documented, and current mathematical defenses offer only partial protection [13], [27].

• Dataset fidelity and evaluation rigor (established challenge): Public intrusion and malware datasets often contain artificial traffic flaws, wrong labels, and sample ratio or time biases [9], [20], [21].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

• Concept drift and delayed labels (established challenge): Ongoing changes in normal software and attacker tactics cause static models to lose accuracy quickly without active learning and scheduled retraining [19], [20].
• Practitioner-centered explainability (active research): Security analysts need short, clear, and reliable explanations to verify alerts quickly during busy shifts [14], [23].
• Privacy-preserving collaborative learning (early empirical stage): Federated intrusion detection looks promising for sharing threat insights across organizations, but still faces hurdles with uneven data and poisoning attacks [15].
• Scalable deployment and human governance (established practice): Balancing the hardware limits of small edge devices [17] against the high server costs of big-data platforms [3] while keeping human analysts in control remains a tough engineering task.
• Dual-use generative AI and LLMs (emerging area): The UK NCSC [2] expects generative AI to boost reconnaissance and social engineering attacks through 2027—even if fully automated end-to-end attacks remain unlikely in the short term—while NIST [27] outlines prompt-injection and poisoning attacks against generative models. At the same time, Xu et al. [16] highlight promising defensive uses of LLMs in finding software bugs and analyzing malware, though independent field tests measuring how much they actually help defenders are still rare.

C. Open Research Gaps and Future Directions

Fig. 5 connects seven major research gaps found in the literature to practical directions for future work and notes how mature each area is. Closing the gap between high lab test scores and reliable real-world security requires focusing on long-term field studies that respect time order, test under realistic attack rates, and measure how well models fit into daily SOC analyst workflows.

Identified Research Gap	Recommended Investigation Direction	Current Maturity Level
Dataset Fidelity & Label Rigor	Reconstruct flows, remove artifact flaws, evaluate with realistic benign-to-malicious attack ratios [21]	Established Challenge
Temporal Concept Drift & Skew	Time-consistent train-test splits (TESSERACT), continual retraining, delayed feedback modeling [19], [20]	Established Challenge
Adversarial Robustness Testing	Empirical evasion & poisoning testing adhering to NIST AI 100-2e2025; certified defense bounds [13], [27]	Established Challenge
Practitioner-Centered XAI	Task-grounded explanation interfaces tested with active SOC analysts to reduce cognitive fatigue [14], [23]	Active Research
Privacy-Preserving Federated IDS	Decentralized IDS with Byzantine fault tolerance, heterogeneous data normalization & poisoning defense [15]	Early Empirical
Scalable Edge-to-Cloud Governance	Hierarchical edge scoring (KitNET) coupled with cloud big-data lake and strict human oversight [3], [17]	Established Practice
Dual-Use Generative AI & LLMs	Defenses against prompt injection, malicious synthetic code, and rigorous LLM bug auditing evaluation [16], [27]	Emerging Frontier

Fig. 5. Research gaps identified in the reviewed literature mapped to recommended investigation directions and current maturity levels.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

IX. CONCLUSION

This literature review examined thirty-two confirmed sources to see how AI, ML, and DL are used in digital security, what level of performance is backed by solid evidence, and what practical limits affect real-world deployment. Three main conclusions stand out from this review. First, statistical and neural learning models provide real value when analyzing huge volumes of monitoring data, catching unusual behavior and modified threats that slip past static rules and cutting down triage work for analysts in both research prototypes and commercial filters [3], [17], [25], [26]. Second, high scores on benchmark datasets often exaggerate how well models will work in practice unless testing setups remove sample ratio bias, time leakage, and dataset label errors while accounting for rare attack rates and changing traffic [9], [19], [20], [21], [22]. Third, major weaknesses remain unsolved around adversarial manipulation, black-box predictions, data privacy, analyst alert fatigue, and the dual-use risks of generative AI [13], [14], [15], [16], [23], [27]. Because of this, AI and ML work best when built into layered defense systems alongside traditional security rules, ongoing drift checks, and human oversight.

X. PRACTICAL RECOMMENDATIONS

A. Guidelines for Security Researchers

1. Use strict time-ordered train-and-test splits and realistic ratios of safe to malicious events instead of random cross-validation, and report precision, recall, and false-positive rates within realistic daily alert limits rather than accuracy alone [9], [20].
2. Check public benchmark datasets for traffic-generation flaws, flow errors, and wrong ground-truth labels before training models, and state dataset limits clearly [21].
3. Test how well models hold up against realistic adversarial changes using standard frameworks like NIST AI 100-2e2025 [13], [27].
4. Test explanation tools and alert formats through practical studies with working SOC analysts instead of relying only on mathematical proxy scores [14], [23].
5. Carry out independent, long-term field tests of deployed security models, federated learning setups [15], and defensive LLM tools [16].

B. Guidelines for Deploying Organizations

1. Choose model complexity based on the specific security task, how many labeled examples are available, and hardware budgets, testing candidate models on local network data before rolling them out.
2. Set up ongoing data-quality checks, regular retraining schedules, and automated drift monitors so models do not go stale over time [19], [20].
3. Keep human analysts in the loop to approve high-impact actions like isolating hosts, and clearly separate statistical model scores from automatic blocking rules (Fig. 3) [3], [23].
4. Pair learning-based detectors with layered traditional defenses, including strong encryption, least-privilege access controls, signature filters, and regular software updates [30].
5. Protect training pipelines and live models against poisoning and evasion attacks, and manage model risks across their lifecycle using structured guidelines like the NIST AI RMF [27], [31].

ACKNOWLEDGMENT AND DECLARATIONS

Varad Lohakane and Priyanka Ghode express their gratitude to Manisha Wakchaure and the Department of Information Technology at Government Polytechnic, Awasari Khurd, for their academic mentorship and technical support.

Author contributions: Varad Lohakane established the review scope, screened and verified the bibliographic sources, compiled the domain synthesis, and drafted the paper; Priyanka Ghode guided the research design, reviewed the comparative evaluation, and edited the final manuscript.

Generative AI disclosure: An AI tool (Claude by Anthropic) was used to assist with literature search, reference verification, figure creation, and document formatting. The AI system conducted no experimental trials and is not an

author; all human authors checked every claim and citation against the original sources and take full responsibility for the work.

Conflict of interest and funding: The authors report no competing interests and received no outside funding for this work.

Ethical approval: Not required, since this review examines publicly available academic and government publications and involves no human subjects or private data.

REFERENCES

[1] "Standards for security categorization of federal information and information systems," National Institute of Standards and Technology, Gaithersburg, MD, USA, FIPS PUB 199, Feb. 2004.

[2] National Cyber Security Centre, "Impact of AI on cyber threat from now to 2027," NCSC, London, U.K., 2025. [Online]. Available: https://www.ncsc.gov.uk/report/impact-ai-cyber-threat-now-2027

[3] K. Veeramachaneni, I. Arnaldo, A. Cuesta-Infante, V. Korrapati, C. Bassias, and K. Li, "AI2: Training a big data machine to defend," in Proc. 2nd IEEE Int. Conf. Big Data Secur. Cloud, New York, NY, USA, 2016, pp. 49–54.

[4] T. M. Mitchell, Machine Learning. New York, NY, USA: McGraw-Hill, 1997.

[5] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Hoboken, NJ, USA: Pearson, 2021.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[7] I. H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: An overview from machine learning perspective," J. Big Data, vol. 7, Art. no. 41, 2020, doi: 10.1186/s40537-020-00318-5.

[8] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in Proc. IEEE Symp. Secur. Privacy, Oakland, CA, USA, 2010, pp. 305–316.

[9] D. Arp et al., "Dos and don'ts of machine learning in computer security," in Proc. 31st USENIX Secur. Symp., Boston, MA, USA, 2022, pp. 3971–3988.

[10] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," J. Inf. Secur. Appl., vol. 50, Art. no. 102419, 2020, doi: 10.1016/j.jisa.2019.102419.

[11] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," J. Netw. Comput. Appl., vol. 153, Art. no. 102526, 2020, doi: 10.1016/j.jnca.2019.102526.

[12] A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," J. King Saud Univ. Comput. Inf. Sci., vol. 35, no. 2, pp. 590–611, 2023, doi: 10.1016/j.jksuci.2023.01.004.

[13] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, "Adversarial machine learning attacks and defense methods in the cyber security domain," ACM Comput. Surv., vol. 54, no. 5, Art. no. 108, 2021, doi: 10.1145/3453158.

[14] N. Capuano, G. Fenza, V. Loia, and C. Stanzione, "Explainable artificial intelligence in cybersecurity: A survey," IEEE Access, vol. 10, pp. 93575–93600, 2022, doi: 10.1109/ACCESS.2022.3204171.

[15] S. Agrawal et al., "Federated learning for intrusion detection system: Concepts, challenges and future directions," Comput. Commun., vol. 195, pp. 346–361, 2022, doi: 10.1016/j.comcom.2022.09.012.

[16] H. Xu et al., "Large language models for cyber security: A systematic literature review," ACM Trans. Softw. Eng. Methodol., in press, arXiv:2405.04760v5.

[17] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS), San Diego, CA, USA, 2018, doi: 10.14722/ndss.2018.23204.

[18] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), Dallas, TX, USA, 2017, pp. 1285–1298, doi: 10.1145/3133956.3134015.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

[19] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 8, pp. 3784–3797, Aug. 2018, doi: 10.1109/TNNLS.2017.2736643.
[20] F. Pendlebury, F. Pierazzi, R. Jordaney, J. Kinder, and L. Cavallaro, "TESSERACT: Eliminating experimental bias in malware classification across space and time," in Proc. 28th USENIX Secur. Symp., Santa Clara, CA, USA, 2019, pp. 729–746.
[21] G. Engelen, V. Rimmer, and W. Joosen, "Troubleshooting an intrusion detection dataset: The CICIDS2017 case study," in Proc. IEEE Secur. Privacy Workshops (SPW), San Francisco, CA, USA, 2021, pp. 7–12, doi: 10.1109/SPW53761.2021.00009.
[22] S. Axelsson, "The base-rate fallacy and the difficulty of intrusion detection," ACM Trans. Inf. Syst. Secur., vol. 3, no. 3, pp. 186–205, Aug. 2000, doi: 10.1145/357830.357849.
[23] B. A. Alahmadi, L. Axon, and I. Martinovic, "99% false positives: A qualitative study of SOC analysts' perspectives on security alarms," in Proc. 31st USENIX Secur. Symp., Boston, MA, USA, 2022, pp. 2783–2800.
[24] G. Apruzzese et al., "The role of machine learning in cybersecurity," Digit. Threats: Res. Pract., vol. 4, no. 1, Art. no. 8, pp. 8:1–8:38, 2023, doi: 10.1145/3545574.
[25] Microsoft Defender Security Research Team, "Behavior monitoring combined with machine learning spoils a massive Dofoil coin mining campaign," Microsoft Security Blog, Mar. 7, 2018. [Online]. Available: https://cloudblogs.microsoft.com/microsoftsecure/2018/03/07/behavior-monitoring-combined-with-machine-learning-spoils-a-massive-dofoil-coin-mining-campaign
[26] N. Kumaran, "Spam does not bring us joy—Ridding Gmail of 100 million more spam messages with TensorFlow," Google Cloud Blog, Feb. 2019. [Online]. Available: https://workspace.google.com/blog/product-announcements/ridding-gmail-of-100-million-more-spam-messages-with-tensorflow
[27] A. Vassilev, A. Oprea, A. Fordyce, H. Anderson, X. Davies, and M. Hamin, "Adversarial machine learning: A taxonomy and terminology of attacks and mitigations," National Institute of Standards and Technology, Gaithersburg, MD, USA, Rep. NIST AI 100-2e2025, Mar. 2025, doi: 10.6028/NIST.AI.100-2e2025.
[28] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
[29] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Comput. Surv., vol. 41, no. 3, Art. no. 15, Jul. 2009, doi: 10.1145/1541880.1541882.
[30] "The NIST cybersecurity framework (CSF) 2.0," National Institute of Standards and Technology, Gaithersburg, MD, USA, NIST CSWP 29, Feb. 2024.
[31] "Artificial intelligence risk management framework (AI RMF 1.0)," National Institute of Standards and Technology, Gaithersburg, MD, USA, NIST AI 100-1, Jan. 2023.
[32] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," Pattern Recognit., vol. 84, pp. 317–331, 2018, doi: 10.1016/j.patcog.2018.07.023.